МИНИСТЕРСТВО ОБРАЗОВАНИЯ КИРОВСКОЙ ОБЛАСТИ

Кировское областное государственное профессиональное образовательное

бюджетное учреждение

«Слободской колледж педагогики и социальных отношений»

**ОТЧЕТ**

**по производственной практике**

**ПМ01. Разработка модулей программного обеспечения для компьютерных систем**

**Тема: «Разработка программного модуля «Учет пациентов в больнице»**

Студент

Коротких Наталья Михайловна

Группа 21П-1

Специальность 09.02.07 Информационные системы и программирование

Руководитель практики от колледжа:

*Калинин Арсений Олегович*

Руководитель практики от организации:

*\_\_\_\_\_\_\_\_\_\_\_\_Павленко Сергей Михайлович*

подпись

УТВЕРЖДАЮ:

Директор

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 Наименование организации

\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись расшифровка

М. П.

2024-2025 уч. год

**Содержание**

[1. Сведения об организации (базе практики) 3](#_Toc194269191)

[2. Разработка и анализ технического задания 5](#_Toc194269192)

[3. Разработка программного модуля по алгоритму в соответствии техническому заданию 8](#_Toc194269193)

[3.1. Разработка базы данных. 8](#_Toc194269194)

[3.2. Разработка API 10](#_Toc194269195)

[3.3. Разработка библиотеки для работы с данными 15](#_Toc194269196)

[3.4. Разработка настольного приложения 19](#_Toc194269197)

[3.5. Разработка мобильного приложения 24](#_Toc194269198)

[4. Оформление документации на программный модуль 25](#_Toc194269199)

[5. Выполнение отладки и тестирования программного модуля 26](#_Toc194269200)

[6. Рефакторинг и оптимизации программного кода 27](#_Toc194269201)

[7. Разработка программного модуля для мобильного устройства 28](#_Toc194269202)

[Заключение 29](#_Toc194269203)

[ПРИЛОЖЕНИЕ 30](#_Toc194269204)

1. Сведения об организации (базе практики)

Наименование базы практики – ООО «Электрон-СОФТ».

Статус организации – коммерческая, действующая.

Организационно-правовая форма – общества с ограниченной ответственностью.

Адрес – Кировская область, г. Киров, ул. Ленина, д. 68.

ФИО руководителя (директора) – Устюжанинов Вадим Борисович.

Основной вид деятельности:

* разработка компьютерного программного обеспечения.

Дополнительные виды деятельности:

* торговля оптовая компьютерами, периферийными устройствами к компьютерам и программным обеспечением;
* торговля розничная компьютерами, периферийными устройствами к ним и программным обеспечением в специализированных магазинах;
* деятельность консультативная и работы в области компьютерных технологий;
* деятельность по управлению компьютерным оборудованием;
* деятельность, связанная с использованием вычислительной техники и информационных технологий.

Более 28 лет компания «Электрон» занимается внедрением и обслуживанием систем управленческого и бухгалтерского учета торговых, промышленных и бюджетных организаций.

Компания предоставляет полный спектр услуг по автоматизации управления и учета на предприятиях:

* консультации на этапе подбора программного продукта и его демонстрация;
* поставка программного обеспечения;
* внедрение программного обеспечения;
* сопровождение и обновление программного обеспечения;
* информационно-технологическое сопровождение;
* техническое обслуживание компьютеров и сетей.

1. Разработка и анализ технического задания

Прежде чем приступить к разработке программного модуля «Учет пациентов в больнице», был проведен детальный анализ требований и спроектирована архитектура системы. Этот этап включал изучение нормативной базы, определение ключевых функций модуля и создание наглядных диаграмм, которые стали основой для дальнейшей реализации проекта.

На основании изучения предметной области и заявки заказчика определены ключевые требования к функционалу модуля:

Основные функции модуля:

1. Учет пациентов:

* Регистрация новых пациентов (внесение персональных данных: ФИО, дата рождения, контакты).
* Ведение списка пациентов с возможностью поиска и фильтрации.

1. Управление данными:

* Просмотр и редактирование информации о пациентах.
* Отображение текущего статуса пациента (находится в стационаре, выписан и т.д.).

Нефункциональные требования:

1. Надежность:

* Устойчивость к кратковременным сбоям (восстановление работы в течение 5 минут после перебоев питания).

1. Технические условия:

* Минимальные требования к ПК: процессор 1 ГГц, ОЗУ 1 ГБ, HDD 500 МБ.
* Совместимость с ОС Windows 7/8/10/11.

1. Безопасность:

* Ограничение прав пользователей для предотвращения некорректных действий.

Для наглядного представления функционала разработаны:

1. Диаграмма вариантов использования, которая отображает взаимодействие пользователей с системой (Рисунок 1).

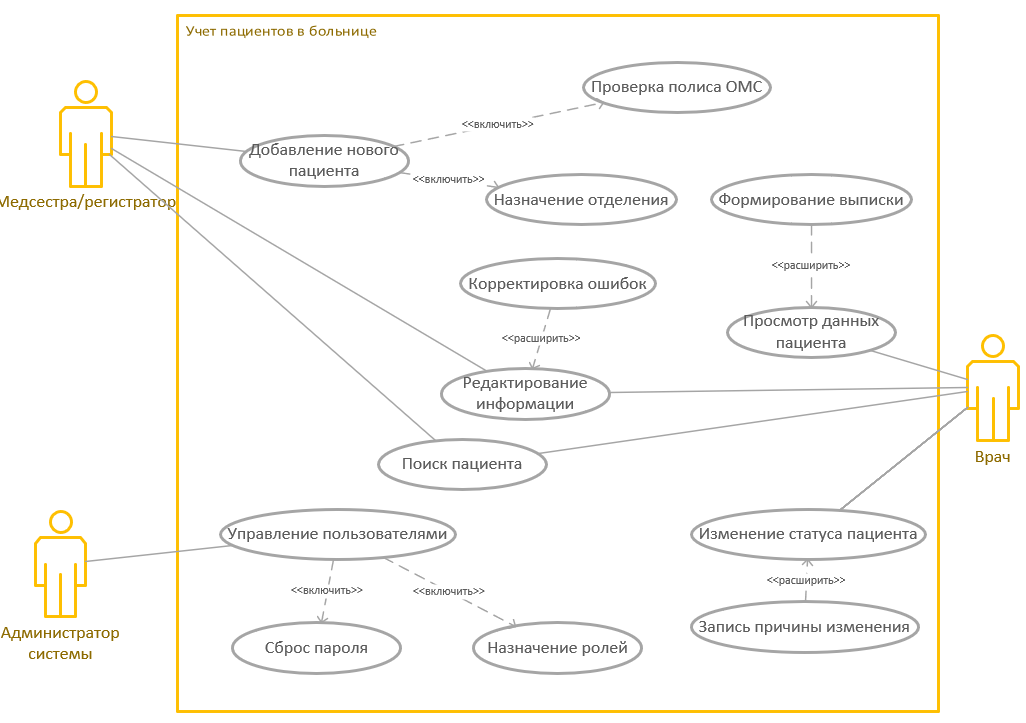


Рисунок 1 – Диаграмма вариантов использования

1. ER-диаграмма – определяет структуру данных (Рисунок 2).

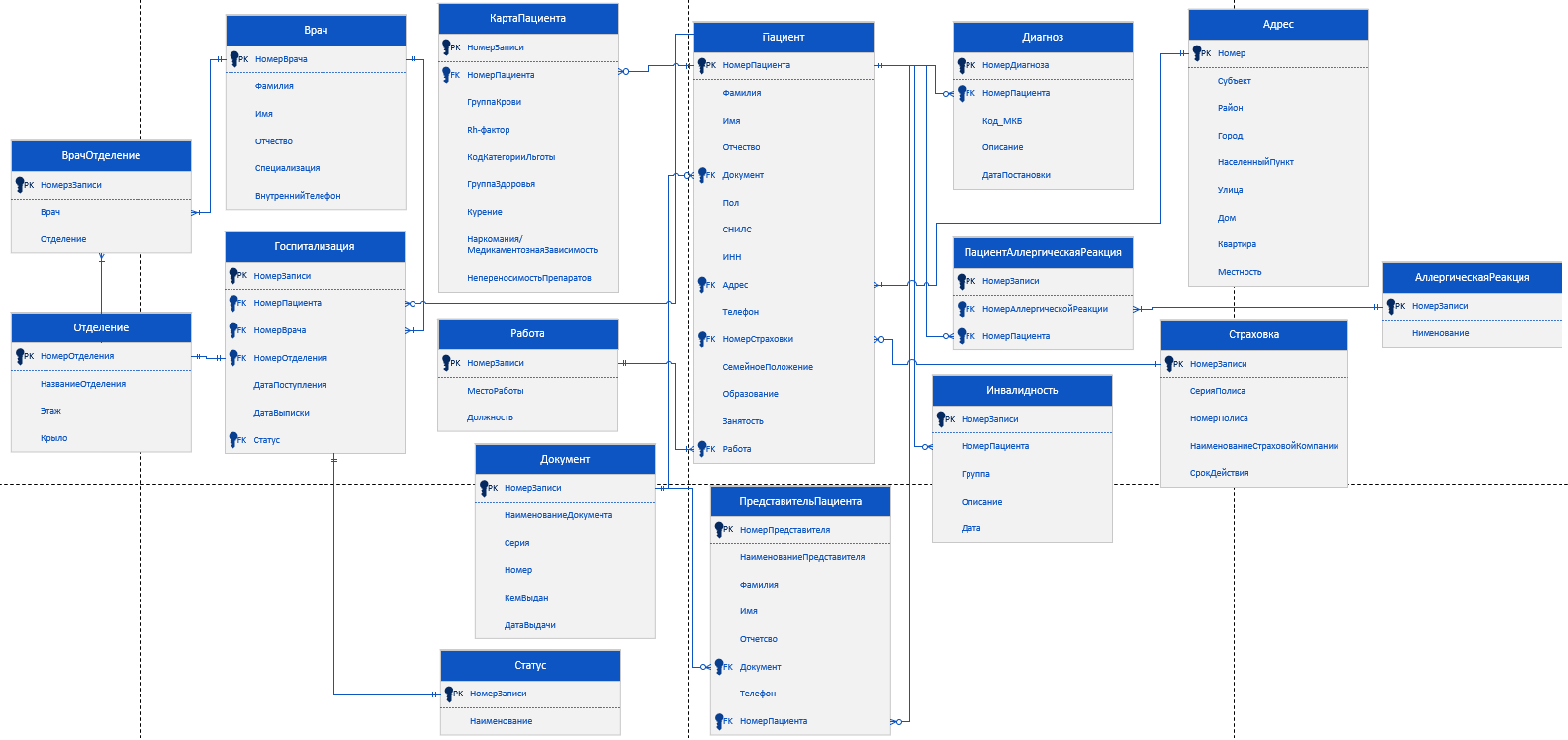


Рисунок 2 – ER-диаграмма

Выбор технологий:

1. Язык программирования: C#.
2. Среда разработки: Microsoft Visual Studio 2022.
3. Хранение данных: Локальная база данных (SQL Server).

Этапы разработки:

1. Анализ требований – определение ключевых функций модуля;
2. Проектирование – создание структуры базы данных и интерфейса;
3. Реализация – написание кода на C#;
4. Тестирование – проверка корректности работы модуля.

В результате анализа технического задания были:

1. Определены основные функции модуля, ориентированные на учет пациентов;
2. Разработаны ключевые диаграммы:

* Диаграмма вариантов использования - для отображения взаимодействия пользователей с системой;
* ER-диаграмма - для проектирования структуры данных.

1. Выбраны технологии разработки (C#, SQL Server);

Анализ предметной области и проектирование системы позволили четко определить логику работы модуля и упростили процесс его последующей реализации. Разработанные диаграммы стали важным инструментом для формализации требований и структуры данных.

1. **Разработка программного модуля по алгоритму в соответствии техническому заданию**

Разработка программного модуля «Учет пациентов в больнице» осуществлялась поэтапно в соответствии с техническим заданием. Основной акцент был сделан на создание надежной системы управления данными пациентов с учетом требований медицинских учреждений.

* 1. Разработка базы данных.

Для хранения и управления медицинскими данными была разработана реляционная база данных на платформе Microsoft SQL Server 2018 v18.12.1. Архитектура базы данных представлена на рис. 3. Она включает 18 взаимосвязанных таблиц, объединенных в логические группы:

1. Основные сущности:

* пациенты (таблица Пациент) с демографической информацией;
* медицинский персонал (таблица Врач);
* структурные подразделения (таблица Отделение);
* пользователи системы (таблица Пользователи) с аутентификационными данными.

1. Медицинские данные:

* карты пациентов (таблица КартаПациента);
* диагнозы (таблица Диагноз);
* аллергические реакции (таблицы АллергическаяРеакция и ПациентАллергическаяРеакция);
* инвалидность (таблица Инвалидность).

1. Административные данные:

* госпитализации (таблица Госпитализация);
* привязка врачей к отделениям (таблица ВрачОтделение);
* статусы пациентов (таблица Статус).

1. Справочная информация:

* адреса (таблица Адрес);
* документы (таблица Документ);
* страховые полисы (таблица Страховка);
* места работы (таблица Работа);
* представители пациентов (таблица ПредставительПациента).

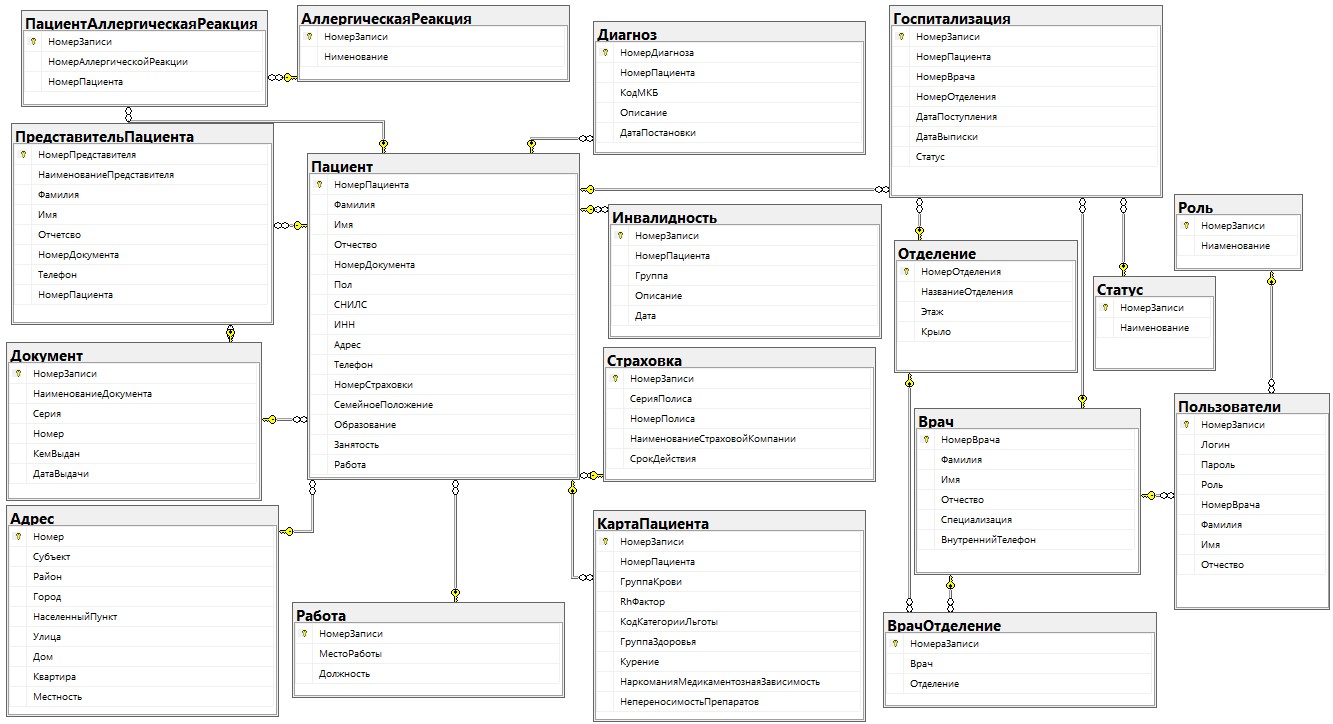


Рисунок 3 – Схема базы данных

Особенностью разработанной структуры базы данных является:

1. реализация системы ролевого доступа с четырьмя уровнями привилегий:

* Администратор (полный доступ);
* Врач (доступ к медицинским данным);
* Медсестра (ограниченный доступ к данным пациентов);
* Регистратор (доступ к регистрационным данным).

1. оптимизированные связи между таблицами через систему внешних ключей;
2. нормализованная структура, минимизирующая избыточность данных;
3. поддержка целостности данных на уровне базы данных;

Для обеспечения безопасности реализована:

1. таблица Пользователи с хранением учетных данных;
2. ограничение доступа к конфиденциальным данным.

Производительность базы данных обеспечена:

1. продуманной структурой отношений между таблицами;
2. минимизацией избыточных данных;
3. эффективными типами данных для каждого поля.

Данная структура позволяет гибко управлять медицинскими данными, обеспечивая при этом необходимый уровень безопасности и соответствие требованиям законодательства о защите персональных данных.

* 1. Разработка API

API системы было разработано на платформе ASP.NET Web API (.NET Framework) с использованием модели ADO.NET Entity Framework для взаимодействия с базой данных.

1. Модель данных и взаимодействие с базой данных.

* Entity Framework (Database-First):
* Модель данных создана с использованием EDMX-дизайнера в Visual Studio.
* Автоматически сгенерированы сущности (классы) для всех таблиц базы данных.
* Конфигурация подключения:
* Строка подключения к базе данных настроена в файле Web.config (Рисунок 4) с использованием провайдера SQL Server.

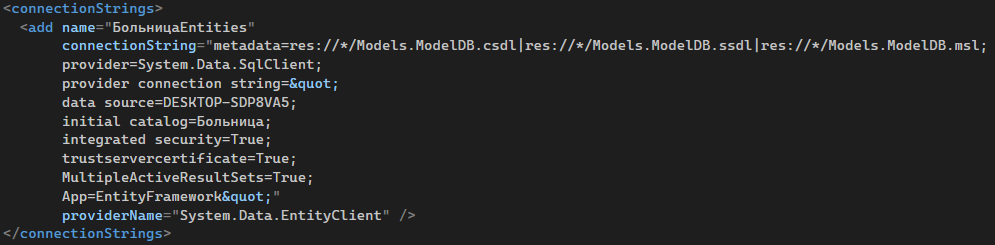


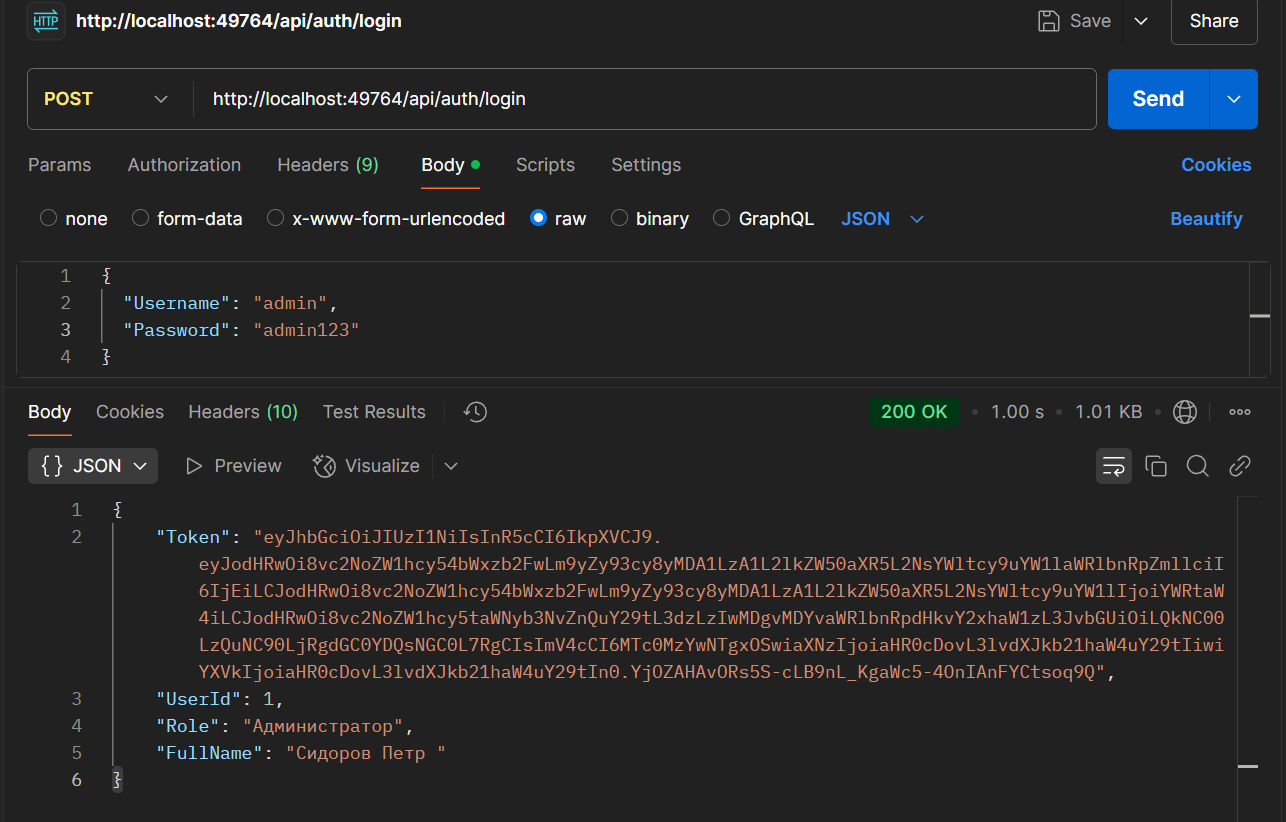
Рисунок 4 – Конфигурация подключения к базе данных

1. Реализация контроллеров и методов API.

API поддерживает RESTful-архитектуру и обеспечивает безопасность через JWT-аутентификацию.

* Авторизация:
* Метод: POST /api/auth/login

Принимает логин и пароль, возвращает JWT-токен. Пример запроса и ответа представлен на рисунке 5.



…

Рисунок 5 – API

* Работа с пациентами:
* Метод: GET /api/patients

Возвращает список всех пациентов в формате JSON. Пример запроса и ответа представлен на рисунке 6.

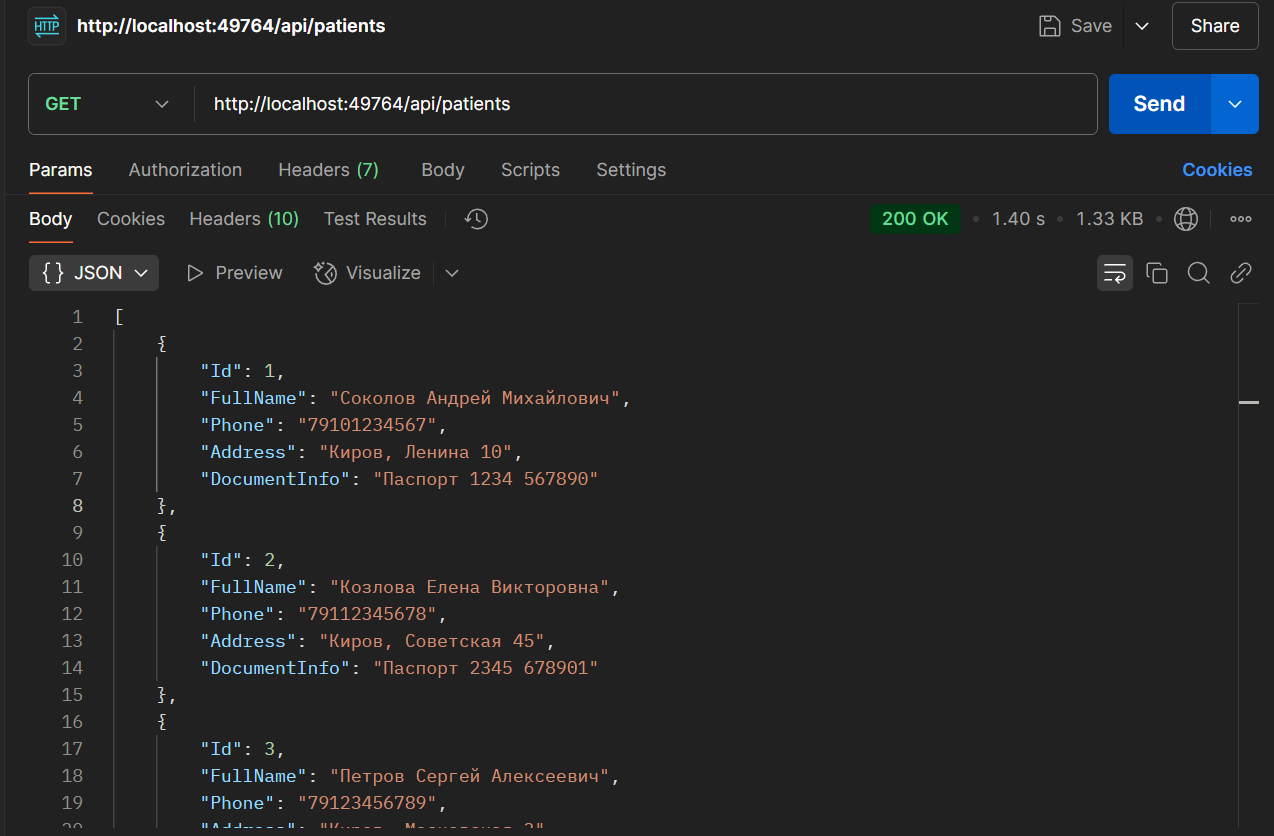


Рисунок 6 – API

* Метод: GET /api/patients/{id}

Возвращает детальную информацию о конкретном пациенте по его идентификатору. Пример запроса и ответа представлен на рисунке 7.

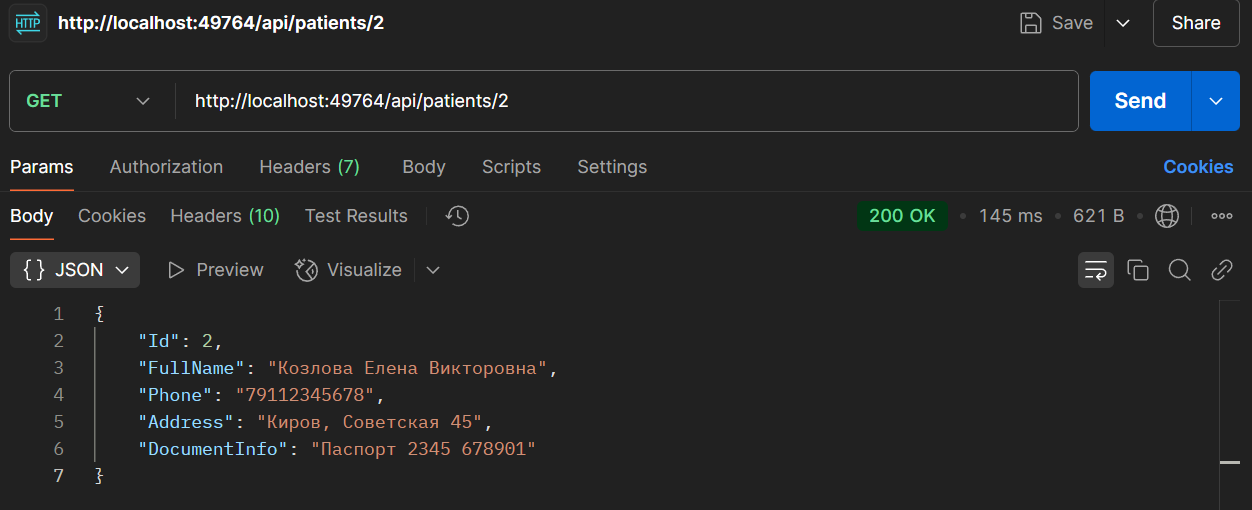


Рисунок 7 – API

* Работа с врачами:
* Метод: GET /api/doctors

Возвращает список всех врачей. Пример запроса и ответа представлен на рисунке 8.

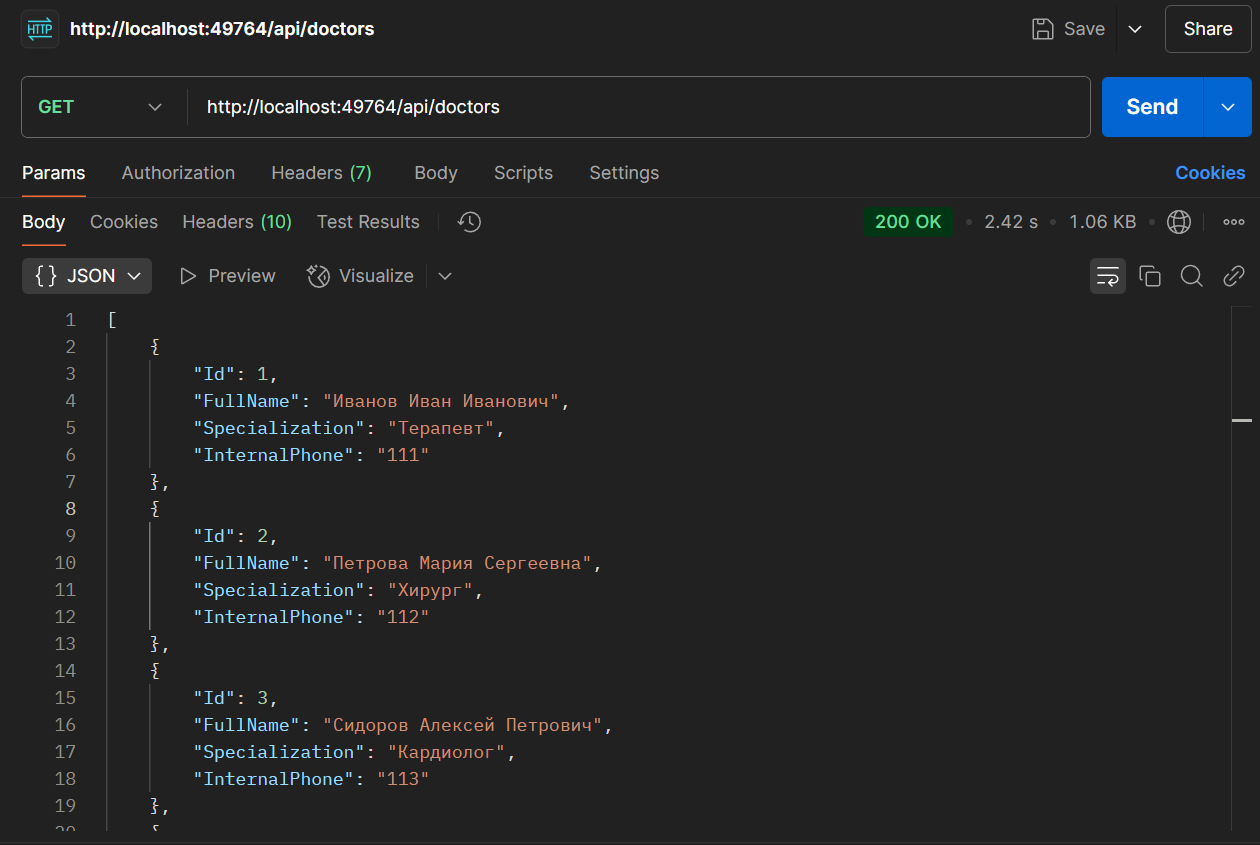


Рисунок 8 - API

* Метод: POST /api/doctors

Создает нового врача на основе переданных данных формате JSON (имя, специализация и др.). Пример запроса и ответа представлен на рисунке 9.

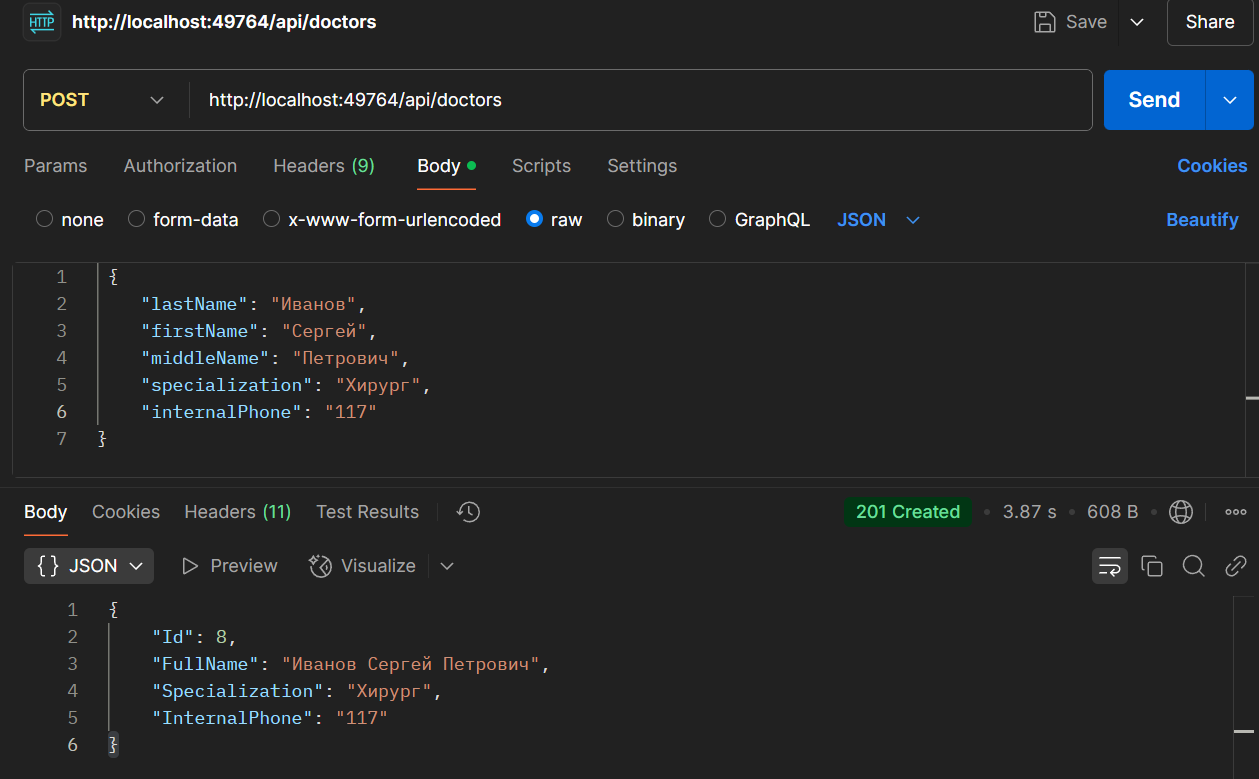


Рисунок 9 – API

1. Документация API системы больницы.

Для каждого конечного пункта API была разработана подробная документация. Документация каждого метода содержит:

* описание назначения API (Рисунок 10);
* параметры запроса (URI и Body) на рис. 11;
* форматы данных (JSON);
* примеры запросов и ответов;
* ограничения на длину строк и обязательные поля (Рисунок 11).

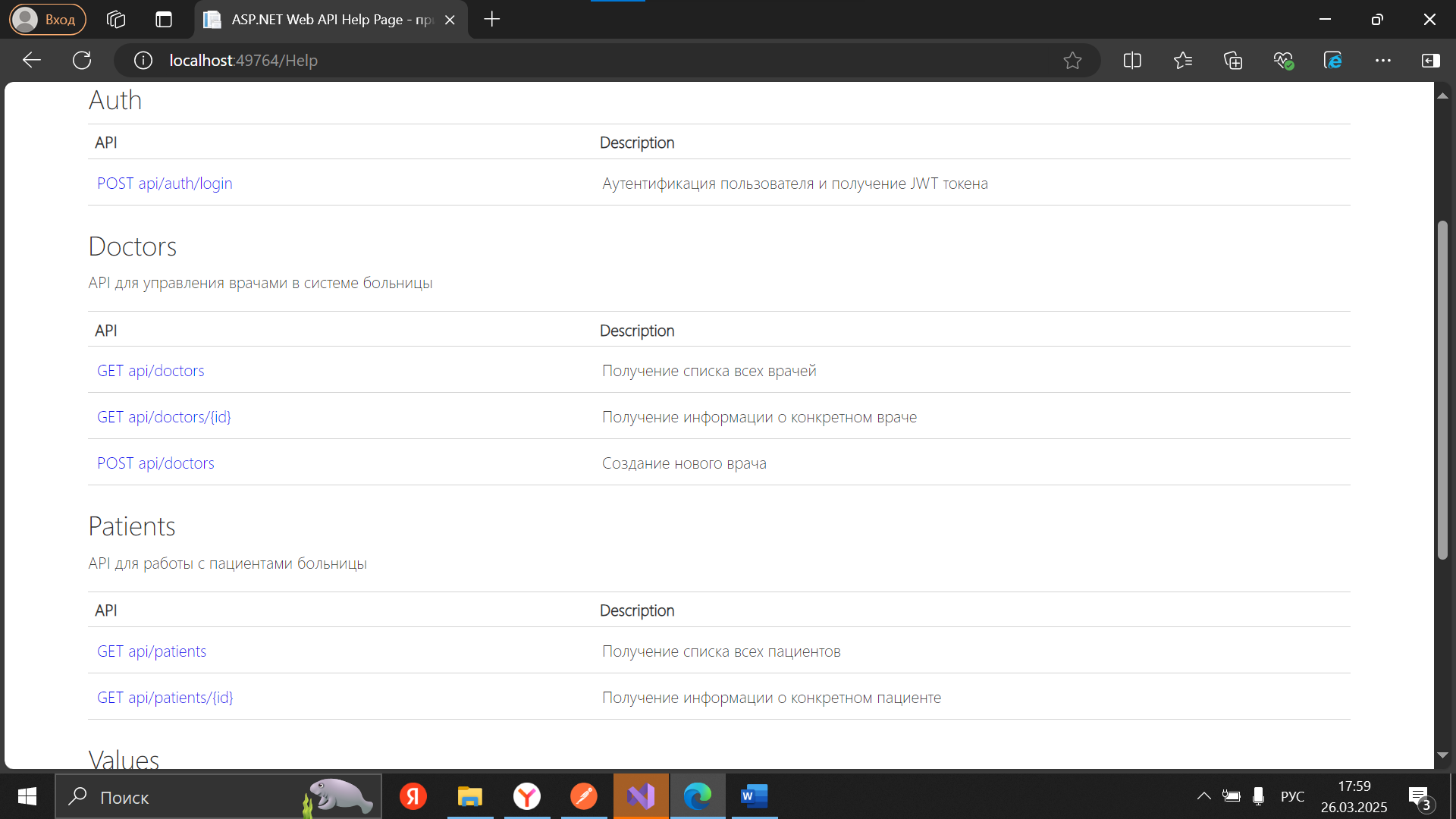


Рисунок 10 – Документация API

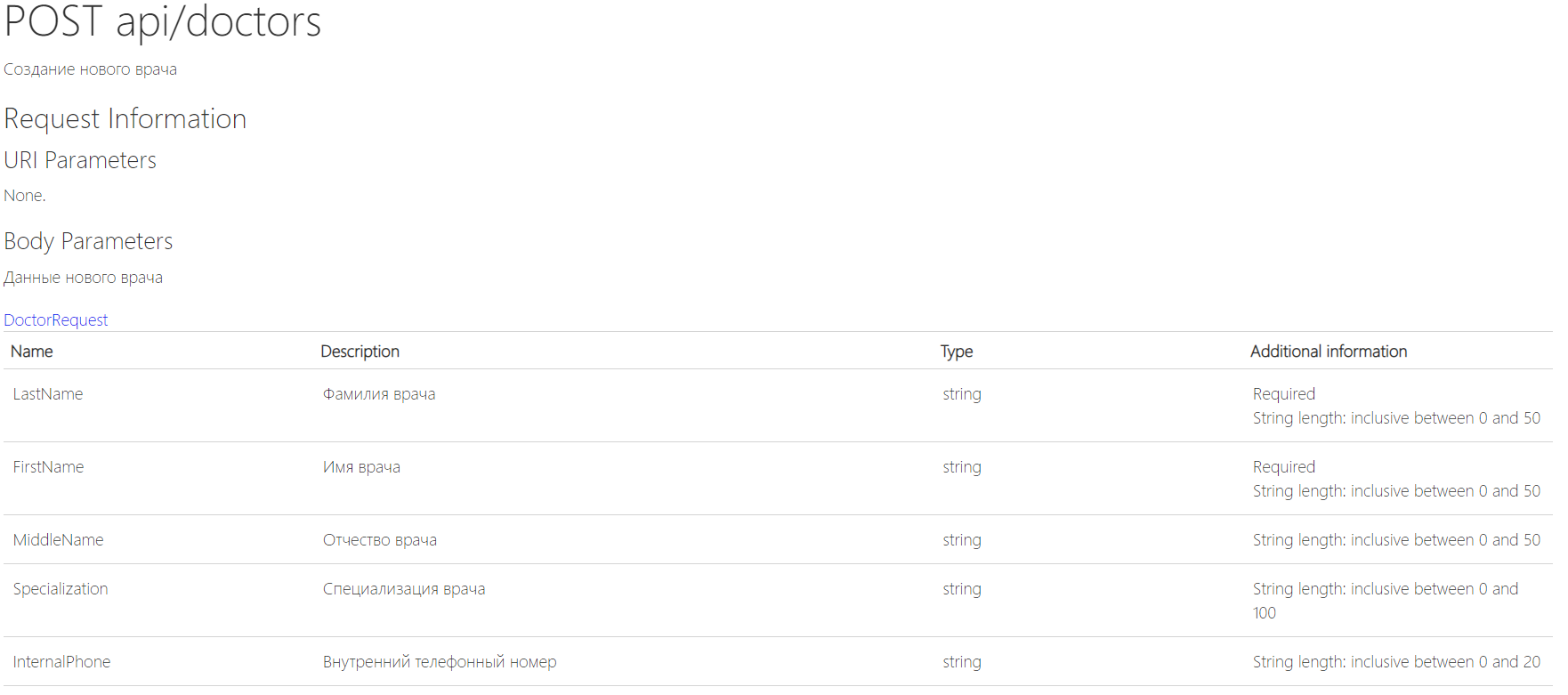


Рисунок 11 - Документация API

* 1. Разработка библиотеки для работы с данными

Библиотека данных реализована в виде Библиотеки классов (.NET Framework) и предоставляет функционал для работы с медицинскими данными пациентов. Основные компоненты:

1. Класс PatientDataService.

Содержит методы для взаимодействия с базой данных:

* Инициализация подключения (Рисунок 12).
* Методы работы с данными:
* GetPatientCount() – получение общего количества пациентов (Рисунок 12).

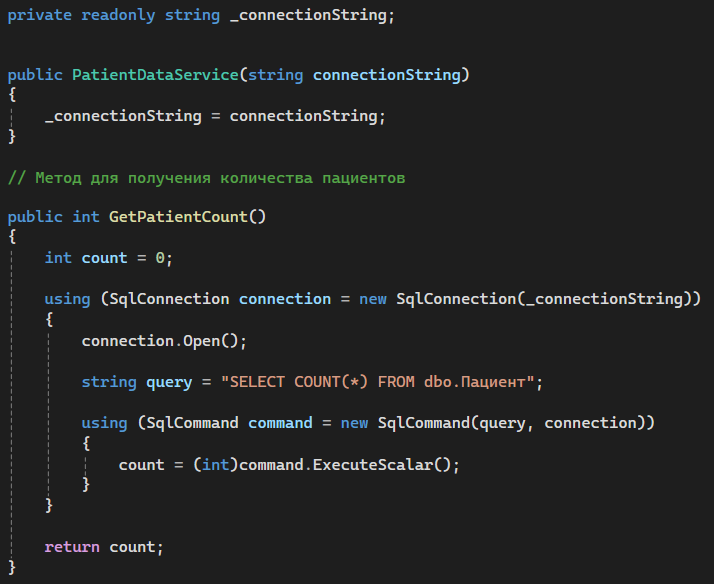


Рисунок 12 – Код библиотеки

* GetMalePatientCount() – получение количества пациентов мужского пола (Рисунок 13).

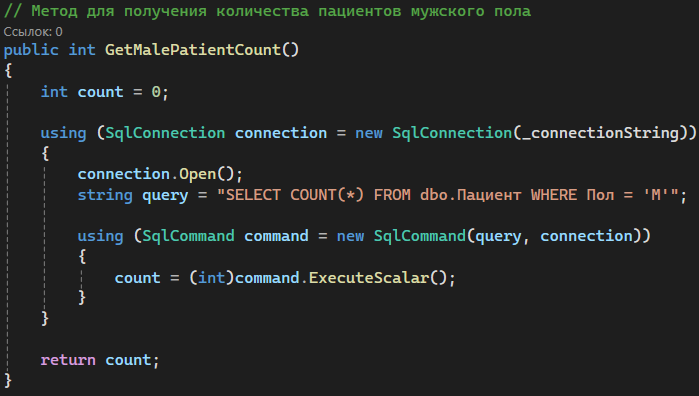


Рисунок 13 – Код библиотеки

* GetFemalePatientCount() – получение количества пациентов женского пола (Рисунок 14).



Рисунок 14 – Код библиотеки

* GetAverageAge() – расчет среднего возраста пациентов (Рисунок 15).

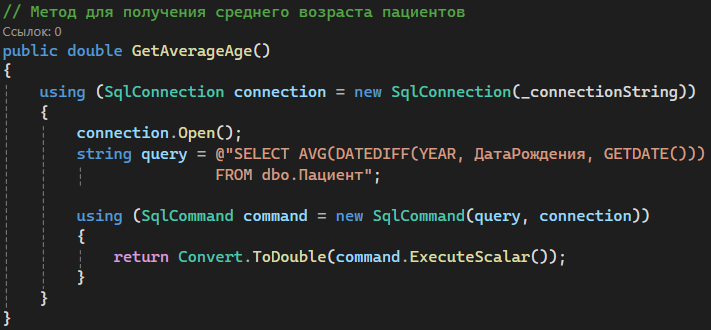


Рисунок 15 – Код библиотеки

* GetPatientsByAgeGroups() – получение количества пациентов по возрастным группам.
* GetPatientsBornInYear(int year) – поиск пациентов, родившихся в определенный год (Рисунок 16).

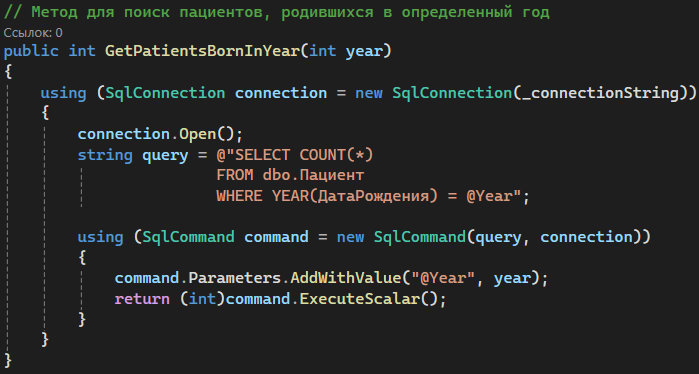


Рисунок 16 – Код библиотеки

* GetBirthdayPersonsThisMonth() – получение списка именинников текущего месяца.

1. Консольное приложение для тестирования.

Для проверки функциональности библиотеки разработано Консольное приложение (.NET Framework), которое:

* Подключается к БД через PatientDataService.

Строка подключения представлена на рисунке 17.

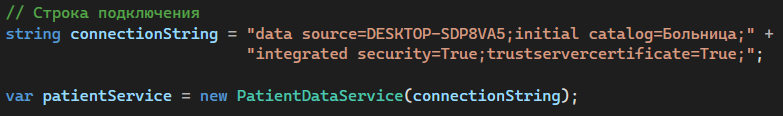


Рисунок 17 – Код консольного приложения

* Выводит статистику в удобочитаемом формате.

Код для проверки метода GetMalePatientCount() представлен на рисунке .

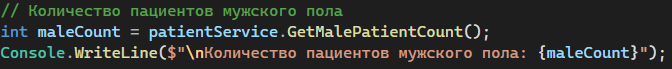


Рисунок – Код консольного приложения

Код для проверки метода GetPatientsByAgeGroups() представлен на рисунке .

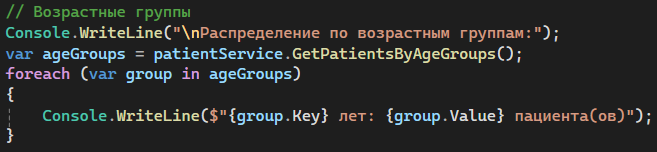


Рисунок -

Код для проверки метода GetPatientsBornInYear(int year) представлен на рисунке .

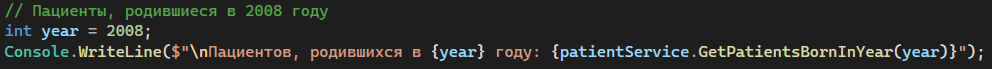


Рисунок -

Код для проверки метода GetBirthdayPersonsThisMonth() представлен на рисунке .

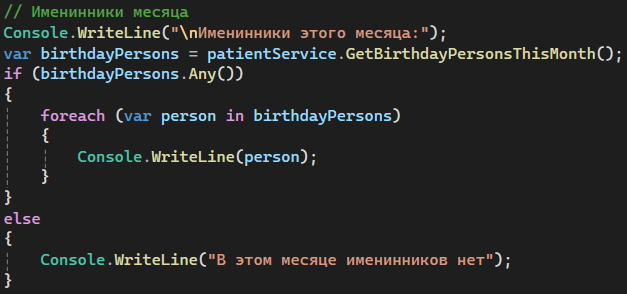


Рисунок -

Результат работы библиотеки представлен на рис. .

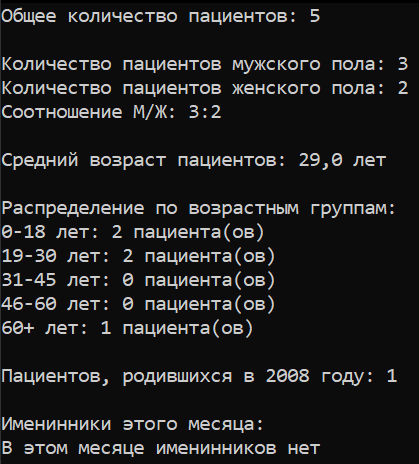


Рисунок – Результат работы библиотеки

* 1. Разработка настольного приложения

Настольное приложение разработано с использованием технологии WPF (Windows Presentation Foundation) на платформе .NET Framework 4.8. Приложение реализует многооконный интерфейс с разделением функционала по ролям пользователей.

1. Модули для разных ролей:

* Администратор:
* Управление пользователями.
* Медсестра:
* Регистрация новых пациентов;
* Обновление основных данных.

Основные окна приложения:

1. Окно списка пациентов с поиском (фамилия, имя, отчество, дата рождения).
2. Форма редактирования данных пациента.

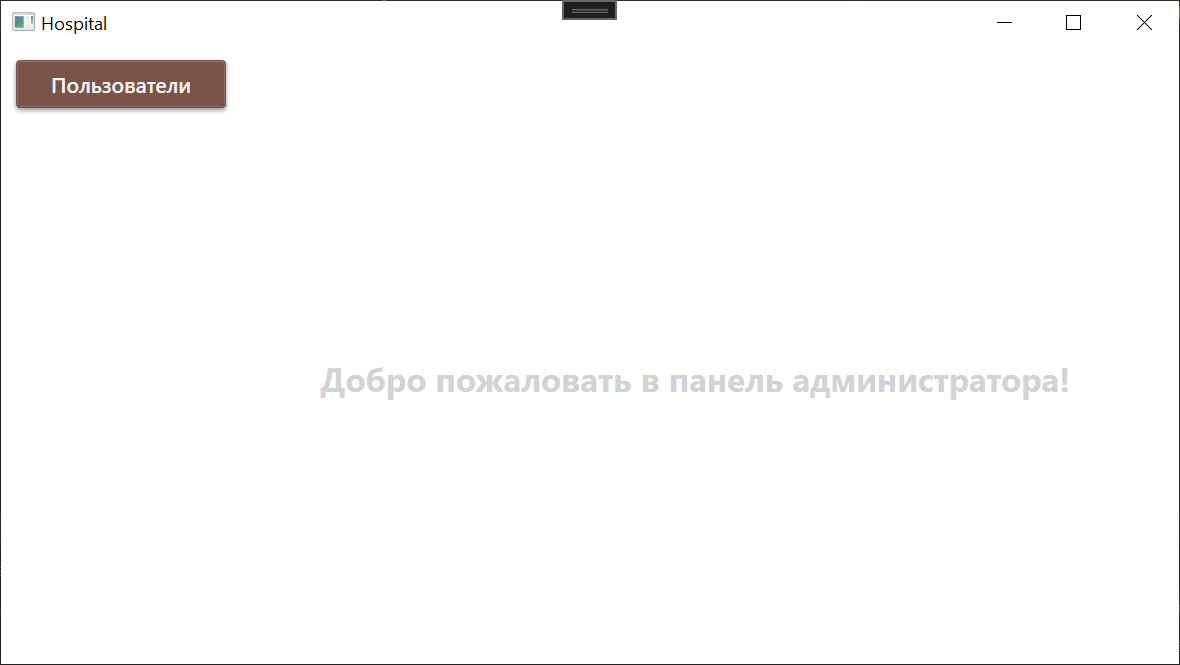


Рисунок – Окно администратора

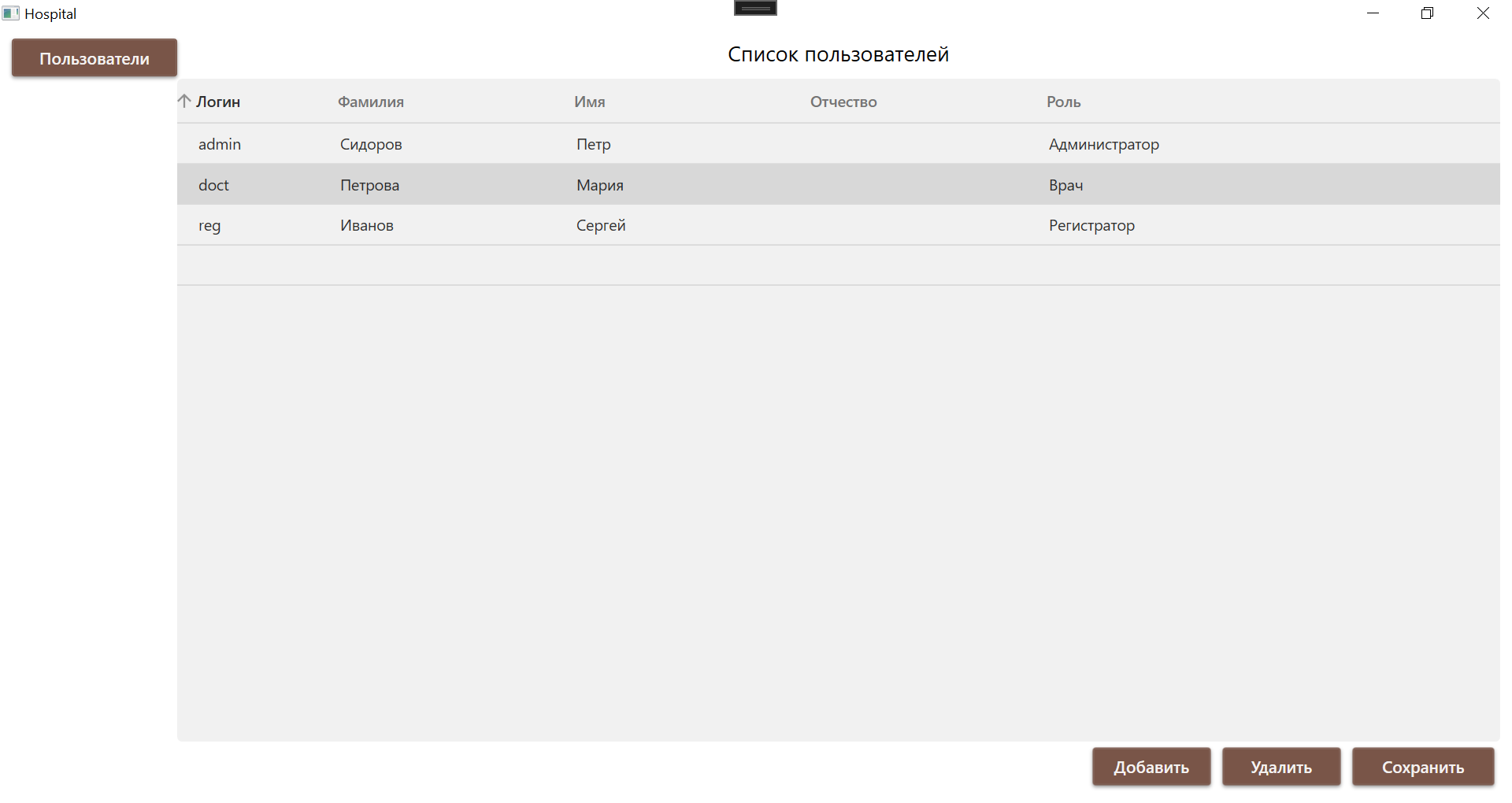


Рисунок – Окно администратора

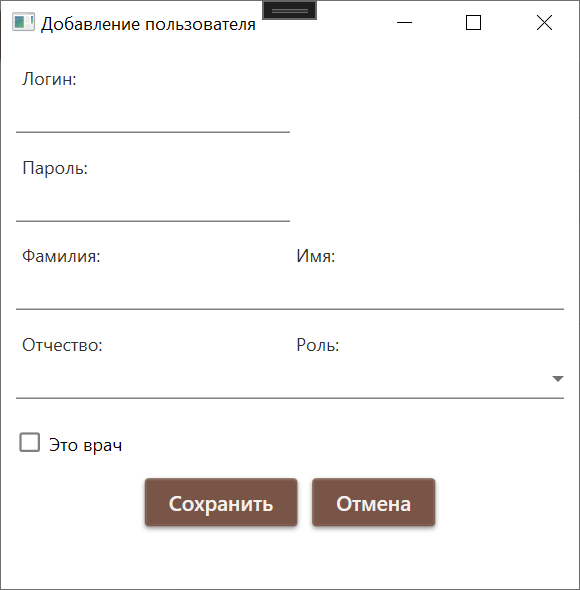


Рисунок – Окно добавления нового пользователя

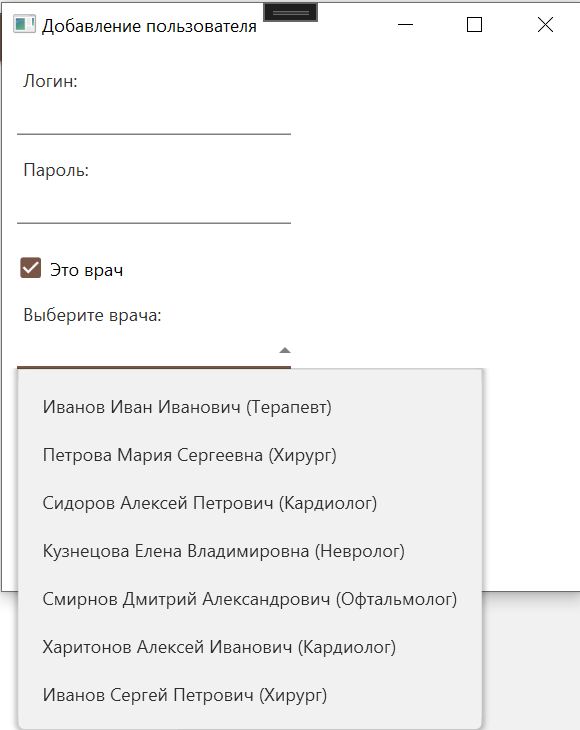


Рисунок – Окно добавления нового пользователя

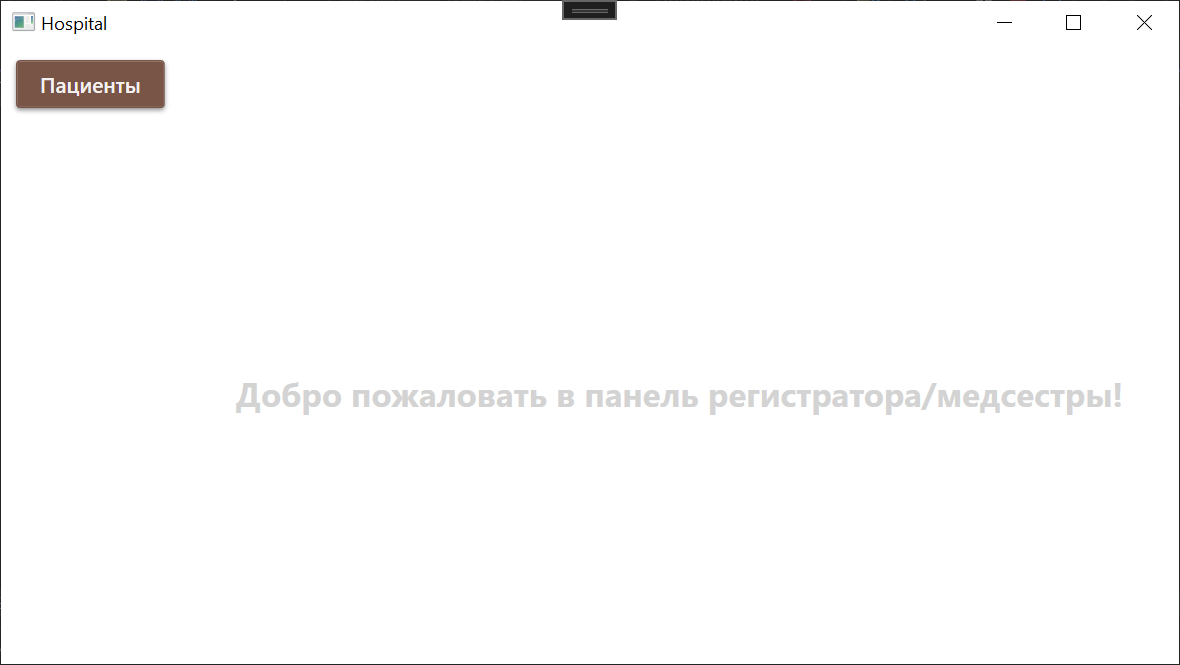


Рисунок – Окно регистратора/медсестры

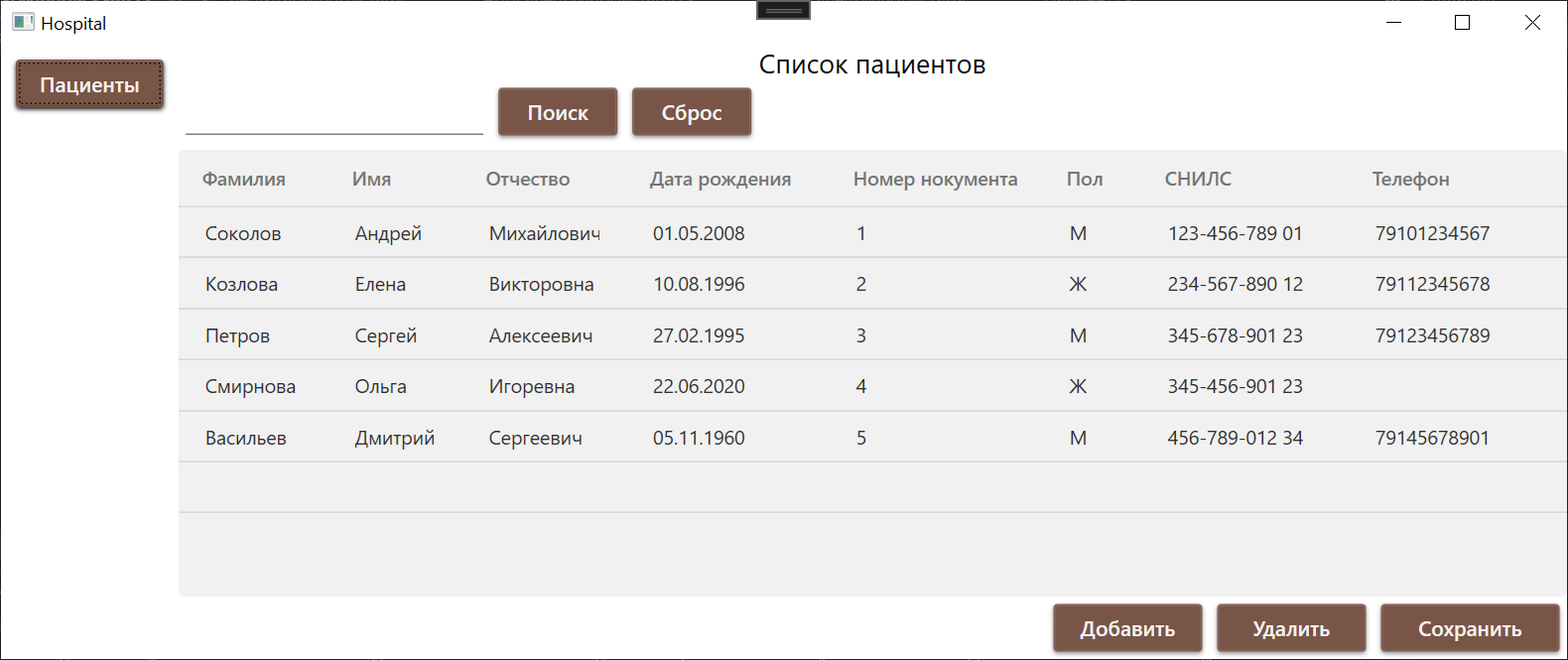


Рисунок – Список пациентов

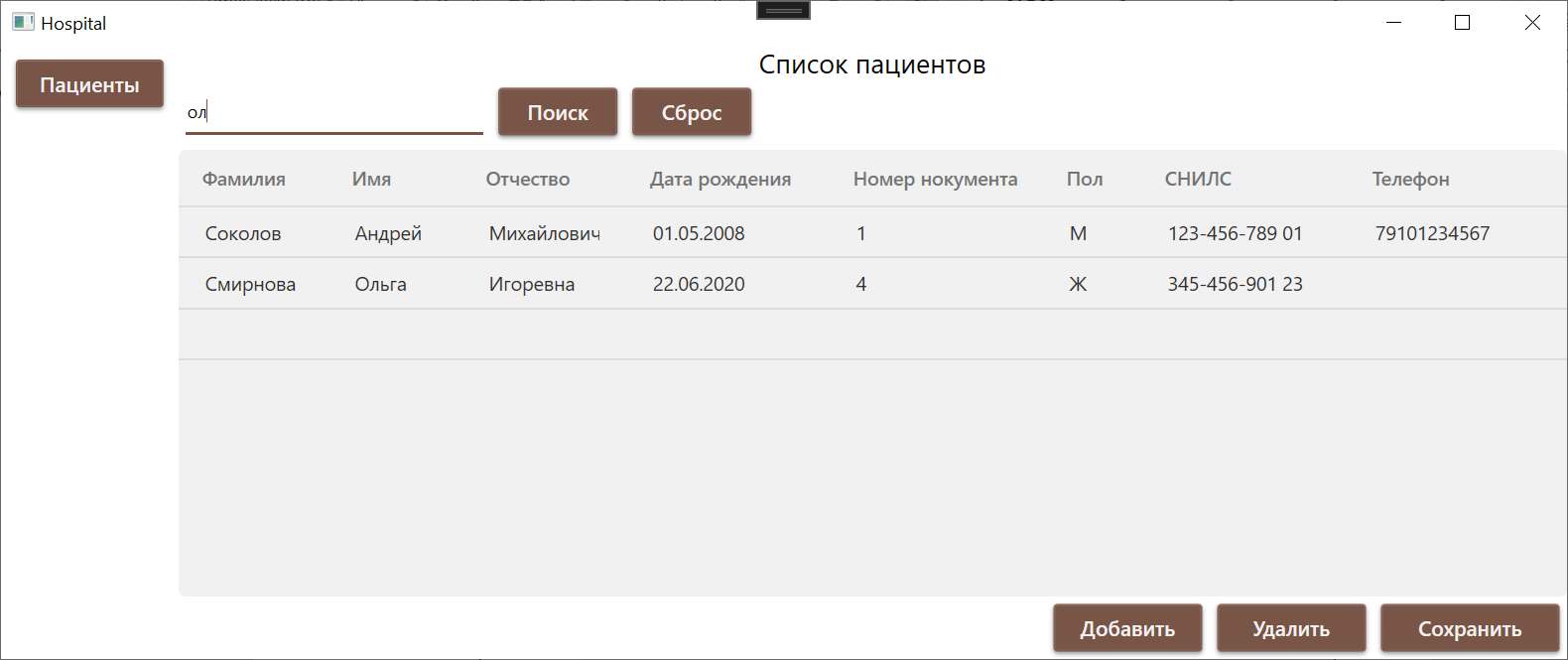


Рисунок – Реализация поиска

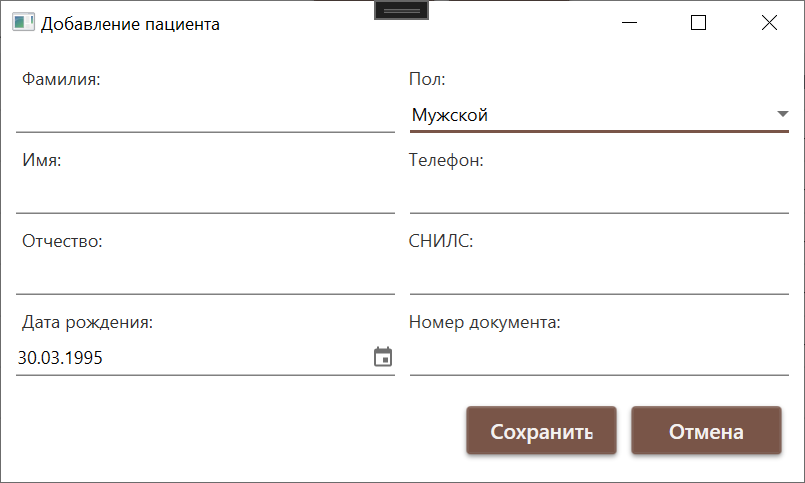


Рисунок – Добавление нового пациента

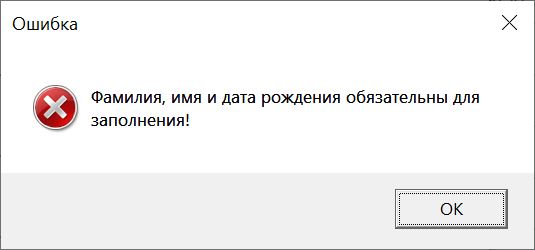


Рисунок – Сообщение об ошибке

Особенности реализации окна для регистратора/медсестры:

1. Валидация данных:

* проверка обязательных полей (фамилия, имя, дата рождения);
* проверка корректности ввода.

1. Удобство использования:

* установлены значения по умолчанию для даты и пола;
* четкие сообщения об ошибках;
* подтверждение удаления.

1. Интеграция с базой данных:

* корректное сохранение всех данных пациента;
* Обновление списка после добавления/удаления;

1. Обработка ошибок:

* перехват и обработка исключений при работе с БД;
* информативные сообщения об ошибках (Рисунок).

Это решение предоставляет полный функционал для работы с пациентами, включая поиск, добавление, удаление и редактирование.

* 1. Разработка мобильного приложения

Мобильное приложение (Xamarin.Forms) предназначено для быстрого доступа к данным:

Основной функционал:

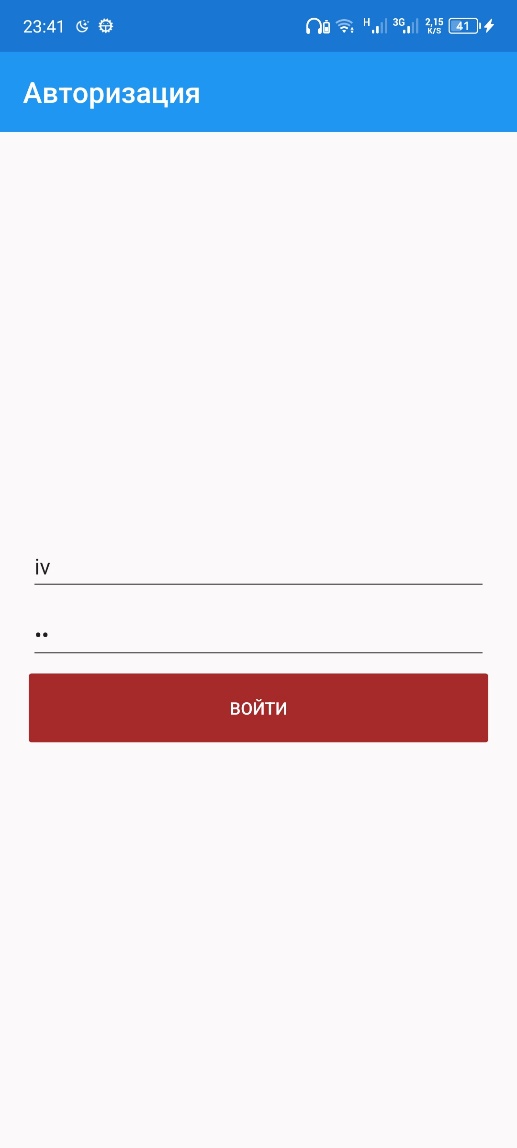


Рисунок – Окно авторизации

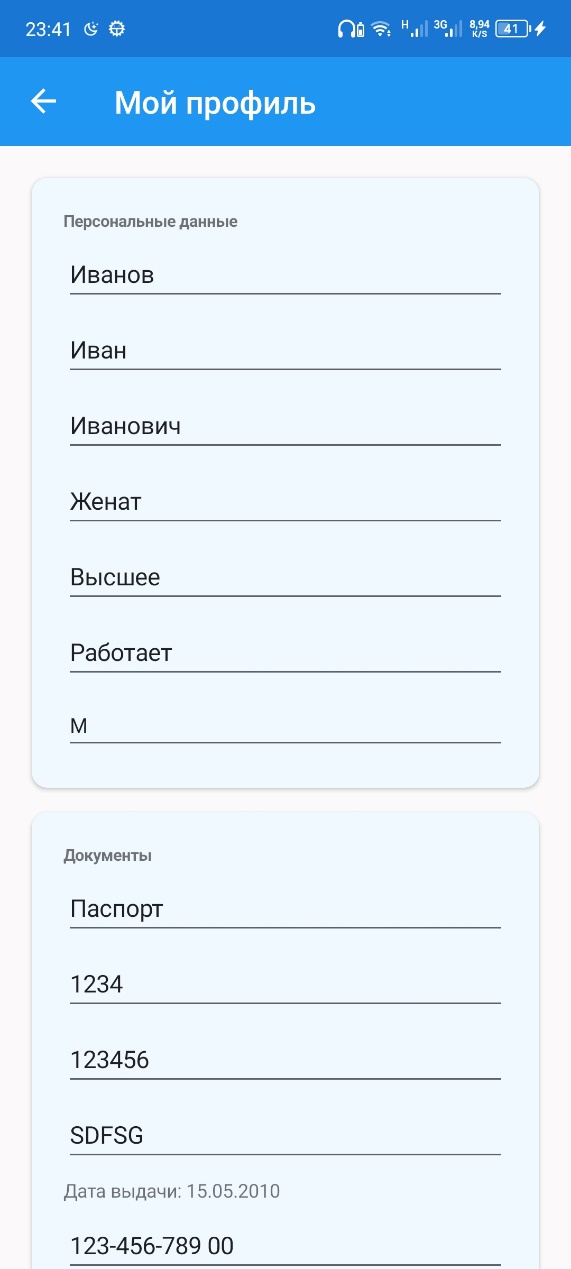


Рисунок – Главное окно

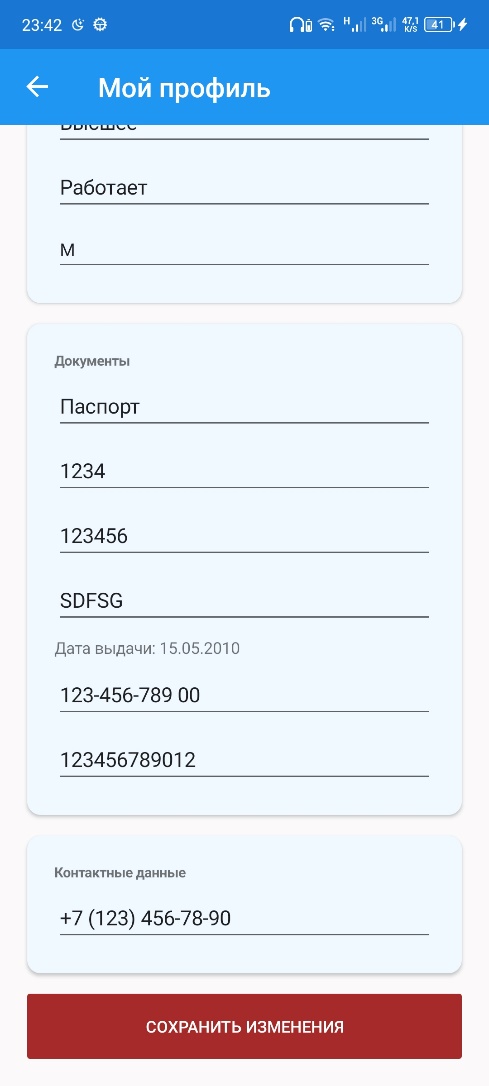


Рисунок – Главное окно

Разработка велась с учетом требований ГОСТ Р 52636-2006 и Федерального закона №323-ФЗ, что обеспечило соответствие системы нормативным требованиям к работе с медицинскими данными.

1. Оформление документации на программный модуль
2. Выполнение отладки и тестирования программного модуля
3. Рефакторинг и оптимизации программного кода
4. Разработка программного модуля для мобильного устройства

Заключение

ПРИЛОЖЕНИЕ

Приложения к отчету:

ссылка на репозиторий отчет в электронном виде, презентация для выступления, проект программного модуля, проект мобильного приложения и др. материалы.

<https://github.com/Natasha975/Hospital>